PATENT ABSTRACTS OF JAPAN

(11)Publication number :   09-034708

(43)Date of publication of application : 07.02.1997

(51)Int.Cl.   G06F   9/44

(21)Application number : 07-179820   (71)Applicant : MATSUSHITA ELECTRIC IND CO LTD

(22)Date of filing :   17.07.1995   (72)Inventor : SANADA NORIO

   SUMIYA KAZUTOSHI

                                                    YASUTAKE KOICHI

                                                    TANAKA HIROHIKO

(54) METHOD AND DEVICE FOR MANAGING OBJECT

(57)Abstract:

PROBLEM TO BE SOLVED: To facilitate the utilization of a class before a change by remaining all the data of class hierarchies at the point of time when changing the class.

SOLUTION: When the change of the class is requested from a change request input part 102, at a class hierarchy version preparing part 106, the copy of class hierarchies including the class to be an object is prepared and that copy is changed so that the class hierarchies before the change can be remained. At the same time, when the class is changed, at a reference class hierarchy change part 107, the names of class hierarchies stored in a hierarchy information storage part 105 are changed into the names of new class hierarchies prepared by the class hierarchy version preparing part 106. As a result, since the class hierarchies at the point of time when utilizing the former class again can be easily provided, it is possible to guarantee the operation of a program prepared from these class hierarchies.
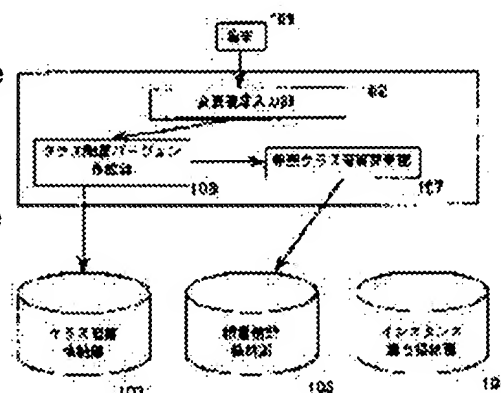
LEGAL STATUS

[Date of request for examination]

[Date of sending the examiner's decision of rejection]

[Kind of final disposal of application other than the examiner's decision of rejection or application

converted registration]

[Date of final disposal for application]

[Patent number]

[Date of registration]

[Number of appeal against examiner's decision of
rejection]

[Date of requesting appeal against examiner's
decision of rejection]

[Date of extinction of right]

* NOTICES *

CLAIMS

[Claim(s)]
[Claim 1] The class hierarchy storing step which stores said class hierarchy in case a class is changed
in the object-oriented system which creates the set of two or more instances from one class hierarchy,
The instance set storing step which stores the set of said instance, The hierarchy storing step which
stores by making into a pair the identifier of a set of an instance, and the identifier of the class
hierarchy who created the set of an instance when the set of said instance is created, The class
hierarchy version creation step which leaves the class hierarchy before modification by creating the
target class hierarchy's copy and changing to a copy when a class is changed, The object management
approach characterized by performing the reference class hierarchy modification step which changes a
class hierarchy's identifier stored at said class information storing step when a class was changed into
a new class hierarchy's identifier created in said class hierarchy version creation section.
[Claim 2] The class hierarchy storing section which stores said class hierarchy in case a class is
changed in the object-oriented system which creates the set of two or more instances from one class
hierarchy, The hierarchy storing section which stores by making into a pair the identifier of a set of an
instance, and the identifier of the class hierarchy who created the set of an instance when the instance
set storing section which stores the set of said instance, and the set of said instance are created, The
class hierarchy version creation section which leaves the class hierarchy before modification by creating
the target class hierarchy's copy and changing to a copy when a class is changed, Object management
equipment characterized by having the reference class hierarchy modification section which changes a
class hierarchy's identifier stored in said class information storing section when a class was changed
into a new class hierarchy's identifier created in said class hierarchy version creation section.

[Translation done.]

* NOTICES *

1.This document has been translated by computer. So the translation may not reflect the original precisely.
2.**** shows the word which can not be translated.
3.In the drawings, any words are not translated.

DETAILED DESCRIPTION

[Detailed Description of the Invention]
[0001]
[Industrial Application] This invention relates to the object management equipment and the object management approach in the system which creates a program using an object-oriented technique.
[0002]
[Description of the Prior Art] In recent years, the system which develops a program using an object-oriented technique is increasing. In object-oriented, a program is reused using the concept of a class and an instance. The contents of the program module used in common by many programs are defined as a class. Moreover, an instance is created as a stereo actually used by the program. Thus, it becomes possible by summarizing a-like definition in a class and performing it in common, to reuse a former code, and it becomes possible to also reflect the correction result of a code in coincidence to all instances.
[0003] There is an approach to carry out class creation of the version high as shown in drawing 5 , whenever it changes a class as an approach of changing the definition of a class in such an object-oriented system. Bibliography: Wong Kim "introduction Thu object oriented database" THE em IT press (Won Kim, Introduction to Object-Oriented Databases, The MIT Press).
[0004]
[Problem(s) to be Solved by the Invention] However, by the conventional approach, since a version is made uniquely, when each class chooses the above two classes in the version of arbitration, these classes cannot check whether it is operating correctly before. So, the instance ****** program created from these classes had the fault that it could not recompense operating correctly.
[0005] Therefore, this invention cancels such a conventional technical problem, and aims at guaranteeing actuation of the program created using the class definition of the version of arbitration.
[0006]
[Means for Solving the Problem] The class hierarchy storing section in which this invention stores a class hierarchy in order to attain this purpose, The hierarchy storing section which stores by making into a pair the identifier of a set of an instance, and the identifier of the class hierarchy who created the set of an instance when the instance set storing section which stores the set of an instance, and the set of an instance are created, The class hierarchy version creation section which leaves the class hierarchy before modification by creating the target class hierarchy's copy and changing to a copy when a class is changed, When a class is changed, it has the configuration equipped with the reference class hierarchy modification section which changes a class hierarchy's identifier stored in the class information storing section into a new class hierarchy's identifier created in said class hierarchy version creation section.
[0007]
[Function] The class hierarchy storing step which stores a class hierarchy in case the object management equipment of this invention changes a class in the object-oriented system which creates the set of two or more instances from one class hierarchy, The hierarchy storing step which stores by making into a pair the identifier of a set of an instance, and the identifier of the class hierarchy who

created the set of an instance when the instance set storing step which stores the set of an instance, and the set of an instance are created, The class hierarchy version creation step which leaves the class hierarchy before modification by creating the target class hierarchy's copy and changing to a copy when a class is changed, By performing the reference class hierarchy modification step which changes a class hierarchy's identifier stored at the class information storing step when a class was changed into a new class hierarchy's identifier created in said class hierarchy version creation section Since it can leave all the class hierarchies at the time of a change of a class being made, the class hierarchy to whom the actuation before modification was guaranteed can be used.

[0008]
[Example] Hereafter, the object management equipment of one example of this invention is explained with reference to a drawing.

[0009] Drawing 1 is the block diagram of the object management equipment in one example of this invention. 101 is a terminal and inputs the identifier of the class which will change from now on. 102 is the change-request input section and is a part which receives the demand of modification to a class. 103 is the class hierarchy storing section and stores two or more class hierarchies. 104 is the instance set storing section and stores two or more sets of the instance created from the class hierarchy. 105 is the hierarchy storing section and stores by making into a pair the identifier of a set of an instance, and the identifier of the class hierarchy who created the set of the instance. 106 is the class hierarchy version creation section, and when the demand of modification of a class comes from the change-request input section 102, it leaves the class hierarchy before modification by creating the copy of the class hierarchy containing the target class, and changing to a copy. 107 is the reference class hierarchy modification section, and when it changes a class, it changes a class hierarchy's identifier stored in the hierarchy storing section 105 into a new class hierarchy's identifier created in the class hierarchy version creation section 106.

[0010] Although the function to specify the class hierarchy who contains the class in fact from the identifier of the class which changes besides this, and the function to input a class hierarchy's identifier created newly are required, since it is not the chief aim of this invention, it omits.

[0011] Actuation of the object management equipment of this example constituted as mentioned above is explained below.

"(Step 201) Class hierarchy storing step"
A class hierarchy's information which the user defined is stored in the class hierarchy storing section 105.

"(Step 202) Instance set storing step"
The set of the instance created from the class hierarchy stored in the class hierarchy storing section 105 is stored in the instance set storing section 104.

"(Step 203) Hierarchy storing step"
In case the set of an instance is stored in the instance set storing section 104 at step 202, the identifier of a set of an instance and the identifier of the class hierarchy who created the set of the instance are stored in the hierarchy storing section 105.

"(Step 204) Modification class information input step"
The change request to the class which changes from the change-request input section 102 is received.

"(Step 205) Class hierarchy version creation step"
The copy of the class hierarchy who contains the class to the class received at step 204 is created, and it changes to a copy.

"(Step 206) Reference class hierarchy modification step"
If the thing same about a class hierarchy's identifier stored at step 203 as the class hierarchy changed at step 205 exists, it will change into the class hierarchy newly created at step 205.
(End)

The example of the object management equipment of this example which operates as mentioned above is explained below. Drawing 2 is the ** type Fig. showing a part of structure of the data stored in the

object management equipment of this example. The data with which 201 is stored in the class hierarchy storing section 103, the data with which 202 is stored in the instance set storing section 104, and 203 express the data stored in the hierarchy storing section 105. In this case, the class hierarchy alpha 1 is stored in the class hierarchy storing section 103, and the instance sets beta1 and beta2 are stored in the instance set storing section 104. Moreover, the data stored in the hierarchy storing section 105 show that the instance sets beta1 and beta2 were created from the class hierarchy alpha 1.

[0012] It considers making now a change to the class a1 contained in the class hierarchy alpha 1. Under the present circumstances, the class hierarchy version creation section 106 creates the copy of alpha 1, and makes it the identifier of alpha 2. And a1 contained in alpha 2 is changed into a2. Then, the reference class modification section 107 changes the data 203 of the components hierarchy storing section 105. This result is shown in <u>drawing 3</u>. They are the data of the class hierarchy storing section 103 after 301 changes, and data of the hierarchy storing section 105 after 302 changes.

[0013] The example of the storing data of the class hierarchy Management Department and the instance set storing section which were created by this example to <u>drawing 4</u> is shown. 401 is the class hierarchy Management Department and 402 is the instance set Management Department.

[0014] As mentioned above, when a class is changed in the object management equipment of this example, it is possible to leave the class hierarchy at the time as it is. For this reason, when creating a program again using the class before modification, it is possible to guarantee actuation of the program created from that class hierarchy by using the class hierarchy containing the version of the class which serves as the purpose from the surviving hierarchy.

[0015]

[Effect of the Invention] When the demand of modification to a class comes as mentioned above according to this invention, a class is changed for the class hierarchy before modification with remnants as it is. For this reason, since the class hierarchy in that time can reappear easily when using a former class again, it is possible to guarantee actuation of the program created from this class hierarchy.

[0016] Therefore, since a user can follow the hysteresis of modification to modification of a class and a former class can be used, the practical effectiveness is large.

[Translation done.]